Proceeding Paper

Enhancement of Consumption Forecasting by Customers' Behavioral Predictability Segregation †

Maria Koshkareva [1,*,‡] and **Anton Kovantsev** [2,*,‡]

1   Department of Digital Transformation, ITMO University, St. Petersburg 197101, Russia
2   National Center for Cognitive Research, ITMO University, St. Petersburg 197101, Russia
*   Correspondence: koshk.mp@gmail.com (M.K.); ankovantcev@itmo.ru (A.K.)
†   Presented at the 9th International Conference on Time Series and Forecasting, Gran Canaria, Spain, 12–14 July 2023.
‡   These authors contributed equally to this work.

Abstract: The easiest approach to customer activity forecasting involves using the whole available and applicable population of customers that a certain data set contains. The drawback of this simple technique is twofold: the set could be too big, and it could contain customers of very different peculiarities, which means that customers whose previous behavior is helpful for the forecast and whose one is not are mixed, and while the first performs a good-quality prediction, the second spoils it by adding noise. Hence, if we could choose the customers with good predictability and put aside the others *"as a shepherd divideth his sheep from the goats" (Matthew 25:32)*, we would solve both problems: less data volume and less noise; the principle is like ancient *"divide et impera"*. In our research, we developed the method of customers separation by predictability and its dynamics with the help of LSTM models. Our research shows that (1) customer separation helps to improve the forecasting quality of the whole population due to the decomposition of all clients' time series, and (2) environmental instability such as pandemics or military action can be leveled out with incremental models.

Keywords: predictability; consumer behavior; incremental learning

1. Introduction

The finance sector has long adopted machine learning techniques in their client behavior analysis (Source code can be found in ref. [1]) to evaluate credit scoring [2], predict customer churn [3], detect fraudulent transactions [4], recommend personalized entities [5], and predict next purchases or trips abroad [6]. Apart from making predictions based on customer behavior, it is wise to detect clients who would be more likely to make such predictions come true, in other words, agents with high predictability. This wisdom is explained by fewer risks when trying out new features and better feedback after targeted recommendations.

Additionally, it may be useful to discard clients with low predictability from available population in quest of higher prediction accuracy. Following the assumption that unpredictable agents add nothing except noise, making predictions for all clients and considering only the ones with high predictability may result in more accurate forecasts. Taking into account that well predicted agents may as well be the first to react to changes and/or have fewer fluctuations in their time series, there is useful information to be extracted from their behavior.

Once highly predictable, an actor may lose this status if their transactional behavior changes due to military-political events, pandemic-related restrictions, or even less influential circumstances. To overcome this, clients' predictability should be evaluated incrementally for shift detection over time.

The primary objective of this paper is to improve forecast quality through financial actor segmentation. In our research, we attempted to do so by extending to the whole population the forecast made for a subset of customers with predictable behavior. Moreover, our incremental learning approach helps to reduce forecast errors caused by environment instability such as pandemics or military action. Source code can be found in ref. [1].

2. Related Works

2.1. Predictability Dynamics

The first mention of dividing financial clients by their predictability can be found in ref. [6], where the authors describe the method of binary client classification based on the predictability of a certain event; clients were divided based on a dataset's median quality metric. The main idea of the method is to perform client segmentation without using a prediction model beforehand. However, a client's predictability is bound to change, which was taken into account in ref. [7]. Not only did the authors use incremental learning techniques for dynamic classification, but they also described the procedure for classifying actors into 32 classes according to the predictability of five chosen transactions. The predictability dynamic can be seen as a transition among classes over time.

Another view on the dynamics of predictability was described in ref. [8], where the dynamic is shown from a predictability quality and model sensitivity perspective. As opposed to previously mentioned works, classification of entities was conducted according to earlier chosen quality metric thresholds, which resulted in five groups.

In this paper, we use the same general idea as in refs. [6,7], including the application of an LSTM model (a recurrent neural network with long short-term memory [9]), but with several alterations: actors are classified based on a forecast quality threshold (similar to [8]) of all transactions; predictions are calculated on all levels, not just the micro- one; the classes of actors are utilized to lower the forecast uncertainty of all clients.

2.2. Incremental Learning

As stated in ref. [10], incremental learning is a learning system that can continuously learn new knowledge and maintain most of the previously learned ones. The main scenarios of incremental learning and its problems can be found in refs. [10,11]. We would do the most basic scenario, also described as fine-tuning to showcase that even simple models can improve upon non-incremental ones when the goal is to overcome a concept drift [12] once critical events occur; the concept drift in our research may appear in a binary classification task on the micro-level when a client stops making transactions for a long time or suddenly starts making them.

According to the first classification of incremental learning scenarios [13], our scenario is domain-incremental learning, according to the second one [14]—"new instances" scenario, because binary classes on the micro-level stay the same, only new samples arrive. The authors in ref. [15] have the same domain scenario but for human state monitoring; they showed that recently developed incremental models have trouble accumulating new knowledge, as opposed to simple models, mainly the ones with replay and cumulative strategies. The authors of another work with the same scenario [16] describe new models' inability to prevent catastrophic forgetting; they perform worse than a replay model. As a result, our most simple model should be able to overcome the concept drift as well as accumulate new information, which is the goal because the model has to adapt and show good forecast quality even after critical events.

3. Materials and Methods

3.1. Dataset Description and Preprocessing

The dataset contains transactional data from our industrial partner: 19,262,668 transactions from 10,000 clients from 1 January 2018 to 15 August 2022. Each transaction is described by a client id, their debit card id, date of a transaction, spent amount, and merchant category code (MCC).

As issues of customers' behavior are of concern in our research, we aggregate payments distributed by MCC to the most obvious groups of consumption interest, which are 'food', 'outfit', 'dwelling', 'health', 'beauty', 'travel', 'kids', 'nonfood', 'telecom', 'fun', and 'money'. The last group is the amount of cash received from ATMs. Moreover, we gather all groups except the one named 'money' into three basic values called 'survival', 'socialisation' and 'self-realisation'. The intuition behind not considering group 'money' is explained by the unpredictable nature of clients' intents once they decide to withdraw money.

The time series for total spending on each of the values is presented in Figure 1. The series were smoothed by a moving average with a 7-day window.
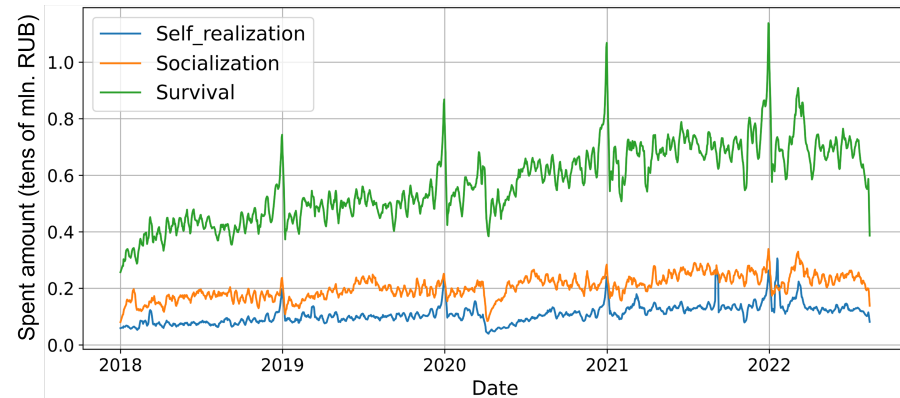

Figure 1. Time series for total expenses for all customers by basic values.

Additionally, to lower the computational cost of the experiment, 3000 of the most active clients were chosen from the available 10,000, where "most active" means the ones with the highest number of transactions through the whole time period.

3.2. Measurement Model for Micro-Level

Before a thorough description, it is worth noting that the idea of predicting the transactional behavior of just one client is usually discarded on the basis of a nearly random series of actions. Consequently, the goal of micro-level forecasting in this research is not going to be the highest possible prediction accuracy. However, the model can find, utilize, and measure recurring patterns in a client behavior sequence, if there are any, which is good enough for the objective of the paper.

The model on the micro-level is used for one client and one transaction group at a time to predict whether any transactions were made in the following days using retrospective transactional data. We consider transactions on a certain day to have occurred if the spent amount is equal to or greater than 10 money units; otherwise, this is a day without any activity. The model calculates predictions for 7 days at once, so it was decided to use the week-based pattern while preparing input data as well, which resulted in 28 days being chosen as input, each with 5 features: whether there were transactions for a given basic value, the sine and cosine of a day of the week number as Equation (1) shows, and the sine and cosine of a month number in a year (2).

$$f_1 = \sin\frac{2\pi D}{7}; f_2 = \cos\frac{2\pi D}{7};\tag{1}$$

where f_1 and f_2 are the day of week features and D is the number of the day in a week (0 being Monday and so on).

$$f_3 = \sin\frac{2\pi M}{12}; f_4 = \cos\frac{2\pi M}{12};\tag{2}$$

where f_3 and f_4 are the month features and M is the number of the month in a year (0 being January).

The model graph can be seen in Figure 2, where LSTM is constructed of two layers with dropout of 20% in between, dropout between dense layers is 10%, and the activation function is leaky rectified linear unit (Leaky ReLU). The graph was drawn by hand to be similar to Keras 'plot_model' function's output because the model was created in PyTorch.
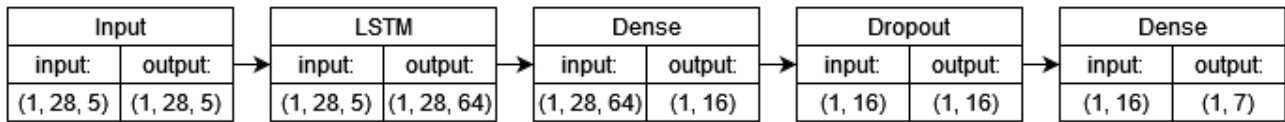
Input			LSTM			Dense			Dropout			Dense	
input:	output:		input:	output:		input:	output:		input:	output:		input:	output:
(1, 28, 5)	(1, 28, 5)		(1, 28, 5)	(1, 28, 64)		(1, 28, 64)	(1, 16)		(1, 16)	(1, 16)		(1, 16)	(1, 7)

Figure 2. Architecture of the predictability measuring neural network.

The loss function is binary cross entropy, optimizer— Adam with a learning rate 5×10^{-4}. Training was conducted with a batch size of 128 for 1000 epochs with early stopping: if a test error is not improving after 10 epochs, the learning rate is lowered by 20%; no improvement after 100 epochs—stop training. Almost every client has a class imbalance: a number of days with completed transactions and days without any. To deal with this problem, the loss function was constructed with a weight parameter, calculated as a ratio of negative instances to positive ones.

3.3. Model for Time Series Forecasting on Meso- and Macro-Level

The model on the meso- and macro-level is used to predict the amount of spent money for a certain class of clients for the next day; therefore, the final layer consists of just one neuron. The model graph can be seen in Figure 3, where LSTM is stateful and is constructed of one layer, dropout between layers is 10%, the activation function is rectified linear unit (ReLU). The graph was created with the use of Keras 'plot_model' function.
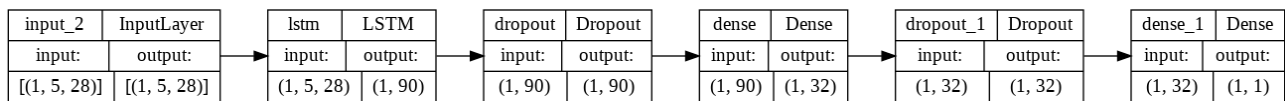
input_2	InputLayer		lstm	LSTM		dropout	Dropout		dense	Dense		dropout_1	Dropout		dense_1	Dense	
input:	output:		input:	output:		input:	output:		input:	output:		input:	output:		input:	output:	
[(1, 5, 28)]	[(1, 5, 28)]		(1, 5, 28)	(1, 90)		(1, 90)	(1, 90)		(1, 90)	(1, 32)		(1, 32)	(1, 32)		(1, 32)	(1, 1)	

Figure 3. Architecture of neural network model for time series forecasting.

Due to the use of stateful LSTM, the batch sizes for training and validation have to match; therefore, its size was decided to be 1. The loss function is mean squared error (MSE), optimizer—Adam with a learning rate 5×10^{-5}. The model was trained for 10 epochs.

These parameters were chosen as optimal after a series of tests. The same was achieved with the measuring neural network on the micro-level.

4. Experiments

4.1. Micro-Level Predictability Measuring

As stated earlier, the objective of the measurement model on the micro-level is to predict whether any transactions will be completed on the following days or not. After encoding time features with sine/cosine transformation, each day is described by 7 features: whether there were transactions in the self-realization group, in the socialization group, in the survival group, sine of a day of the week, corresponding cosine, sine of a month, corresponding cosine.

Available data has critical time periods that should be left out of a training step, which is why the training set consists of a year's worth of days from 1 January 2018 to 31 December 2018, whereas the validation set has days from 1 January 2019 to 15 July 2019.

For further adaptation to client behavior, the model with incremental learning was trained on top of the base one: given 28 days, the model predicts the next seven and then trains on this very data with a higher learning rate (10^{-3}) and without class imbalance techniques to adapt to changes faster.

After training the base model and the incremental one for one basic value group for one client (6 models total for each client), predictions for their transactional behavior are made from 1 January 2019 to 8 August 2022. These forecasts consist of 1 value per day; therefore, the days were merged into weeks with further averaging of the results. The chosen quality metric is the F_1-score (harmonic average of precision and recall); three values for each basic value group were used to calculate a euclidean norm of the three-component vector divided by a square root of three to keep the values in the $[0, 1]$ range.

The forecast quality for an incremental model is used to segment actors into two classes of predictability: clients with a high predictability (F_1-score higher than 0.7) and those with a low one.

4.2. Meso- and Macro-Level Forecasting

As mentioned above, the model at these levels is going to predict the amount of spent money, not just whether there were any transactions on a given day, which is why it is necessary to prepare the data once again: for each class of clients, their sum of spent money per day over the whole time period is smoothed with a moving median with a 21-day window. Only transactions in a group of basic values called "survival" are considered during training and prediction.

The experiment was divided into four parts with two differences: a base model or an incremental one; finding class clients every week; or choosing class clients once. To provide more details regarding the second difference: the first idea consists of choosing corresponding client classes every week, whereas the second one is for working with client classes identified once for the 52nd week because it is one of the last weeks in the relatively calm year 2019. The course of action with the first idea is as follows: find clients in the given class for the current week, predict the spent amount for the following week 1 day at a time, and repeat for each of the available weeks. The plan for the second one differs just in the first step: certain class clients should not be found every week as they are already known and will not change.

All the following experiments were conducted with the same model architecture and hyperparameters to check the hypothesis that it is easier to work with highly predictable clients because they have less noise and fewer unpredictable patterns. Each model was trained in series from the beginning of 2018 until the end of 2019.

Following the objective of the paper, which was to lower the forecast error by client segmentation, the original time series of all actors were compared with the predicted amounts for actors inside a certain class. Taking into account the difference in the number of clients within these classes, the resulting series of forecasts is multiplied by the ratio of median to median, where the first median describes spent money for all clients and the second one for clients in a given class. The chosen quality metric for all forecasts on these levels is mean absolute percentage error (MAPE).

5. Results and Discussion

5.1. Client Segmentation and Incremental Learning on Micro-Level

The quality of predictions given by the base and incremental models on the micro-level can be seen in Figure 4; these are the average results for all clients and all 3 groups of basic values. The first dotted line depicts the week when COVID-19 lockdown was imposed (31 March–4 April 2020), and the second one shows the week when Russia invaded Ukraine (22 February–28 February 2022). It is clear that the incremental model consistently shows higher forecast quality and even has an upward trend in F_1-score.
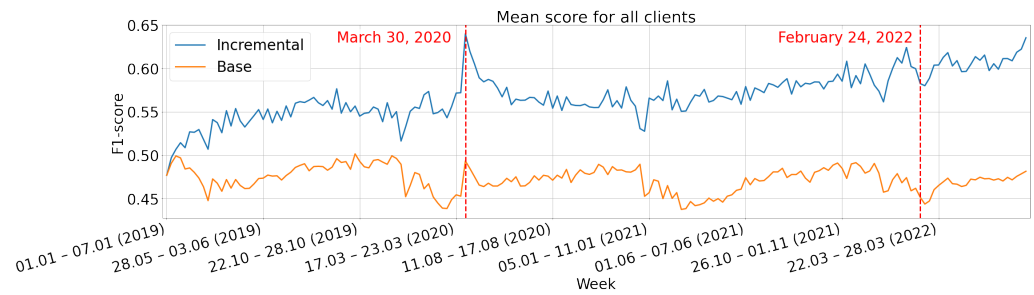
Figure 4. Incremental and base models forecast quality.

Several weeks before the lockdown began, the predictability plummeted, which can be explained by haphazard client behavior in response to the first news regarding deaths from COVID-19 in Russia: some may have begun stocking up on provisions, travelling, and visiting places of interest while it was still possible. The forecast quality for the week of 31 March is the highest, and it is important to remember that the incremental model has high forecast quality when the current week does not differ significantly from previous ones because it adapts to changes and learns from previous instances. The majority of people have been staying at home for several weeks already, so there were few if any transactions made; the incremental model has no problem following such a pattern. The base model does not have methods to adapt to changes, but perhaps some actors returned to their usual behavior seen in the training set after all unusual preparations were completed.

The week when the war began is depicted with decreased forecast quality because the news was less expected than the lockdown imposition, so there was no time to prepare; client behavior became unusual only at the start of the critical period. The given graph also shows reoccurring drops in prediction quality every New Year (1 January), which are explained by expected stress and behavior change due to big holidays.

5.2. Forecast Errors for the Same Class on Meso- and Macro-Level

The first experiment on the meso- and macro-level consisted of measuring forecast quality for the same predictability class in four variations: the trained model was used to predict spent amounts for a given class of clients, which were recalculated each week; the previous model was incrementally trained further; the trained model was used for a given class of clients found on week 52; the previous model was incrementally trained further. The results can be seen in Table 1, where the forecast quality was measured with the help of MAPE and hit probability, which were proposed in ref. [17] and measure the fraction of values in a given range. For easier interpretation, the forecasting error distribution for all experiments is presented in Figure 5.
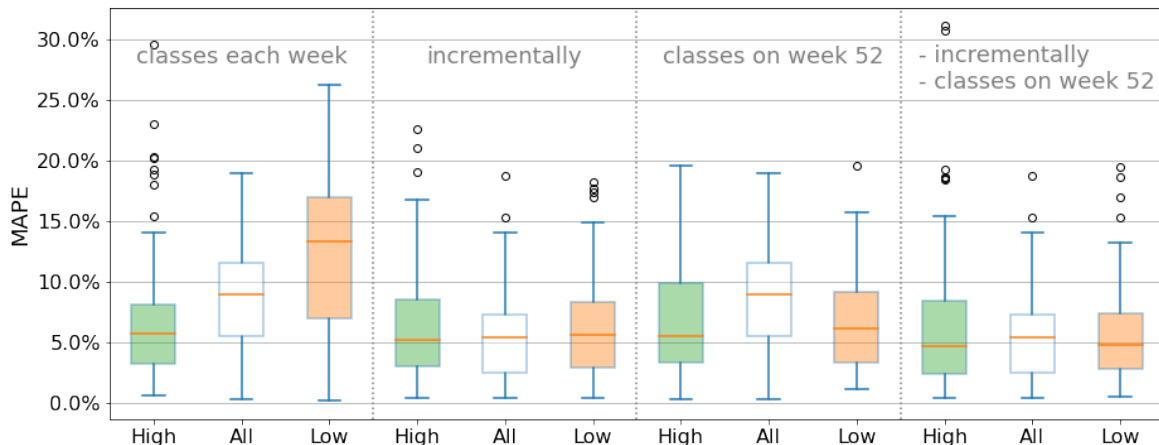

Figure 5. Forecast quality for the same class, where 'High' and 'Low' correspond to clients with high or low predictability.

Table 1. Forecast quality measured for the same class.

Chosen Client Class	Median MAPE	HP: MAPE $\leq$ 5%
Clients in the class were chosen every week		
Clients with high predictability	**5.74%**	**0.422**
All clients	8.97%	0.222
Clients with low predictability	13.32%	0.178
With incremental learning; classes every week		
Clients with high predictability	**5.18%**	**0.481**
All clients	5.38%	0.444
Clients with low predictability	5.62%	0.437
Clients in the class were chosen once on week 52		
Clients with high predictability	**5.53%**	**0.393**
All clients	8.97%	0.222
Clients with low predictability	6.17%	0.333
With incremental learning; classes on week 52		
Clients with high predictability	**4.71%**	**0.548**
All clients	5.38%	0.444
Clients with low predictability	4.82%	0.518

As the results show, clients with high predictability have both the smallest MAPE and the highest hit probability for all cases, which suggests that the model has an easy time training on time series belonging to well predicted agents. The first two cases, where given class clients were found every week, support the hypothesis that clients with low predictability add noise, which results in unpredictable clients having the worst results in these cases. Experiment cases where the class clients were chosen once have a different picture, which can be explained by the following suggestion: the forecast was evaluated across the test period from 1 January 2020 until 15 August 2022; therefore, the clients who were deemed unpredictable on week 52 (from the year 2019) can have predictable patterns in the following years. This assumption supports our idea to use incremental models for classifying clients based on predictability because client's predictability is not constant.

5.3. All Clients Forecast Enhancement on Meso- and Macro-Level

The second experiment's aim was to figure out which class of clients, if any, makes forecasting all clients' time series more accurate. The predictions for given class clients are multiplied by a ratio described in Section 4.2 and compared with the real-time series of all clients with the help of MAPE and hit probability. The predictions are depicted in Figure 6. For easier interpretation, the boxplots for all cases are presented in Figure 7, whereas the numerical results are shown in Table 2.
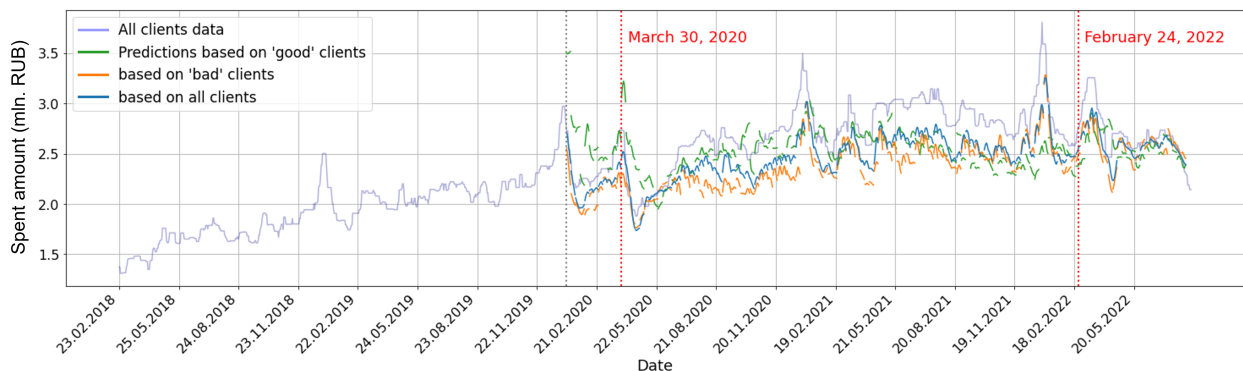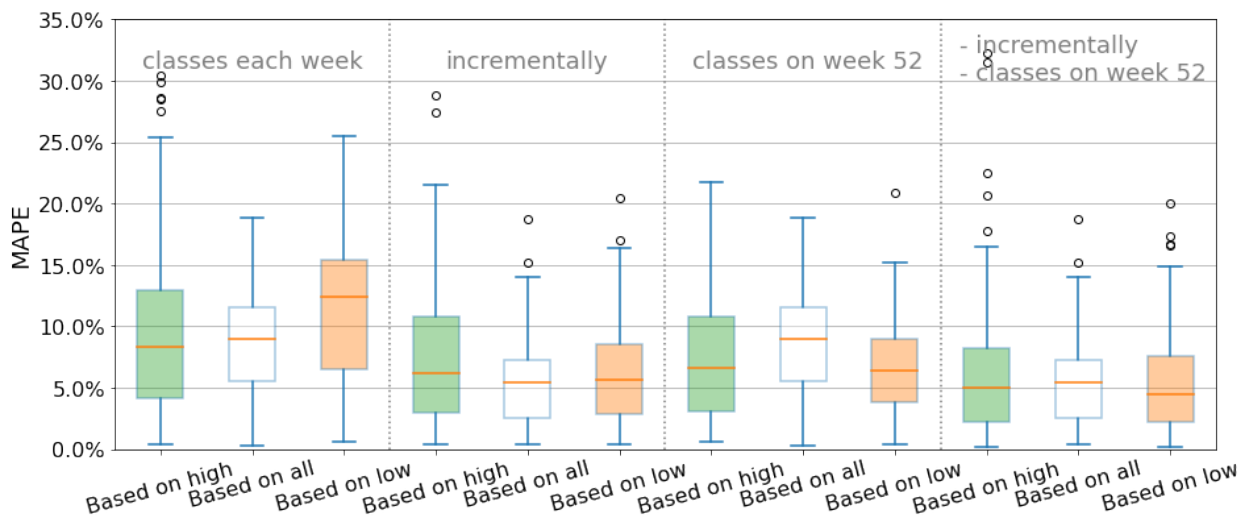

Figure 6. Predictions for all clients.

Figure 7. Forecast quality for all clients, where "Based on high" means based on clients with high predictability.

Table 2. Forecast quality measured for all clients.

Chosen Client Class	Median MAPE	HP: MAPE $\leq$ 5%
Clients in the class were chosen every week		
Based on clients with high predictability	**8.34%**	**0.304**
Based on all clients	8.97%	0.222
Based on clients with low predictability	12.39%	0.148
With incremental learning; classes every week		
Based on clients with high predictability	6.21%	0.370
Based on all clients	**5.38%**	0.444
Based on clients with low predictability	5.67%	**0.459**
Clients in the class were chosen once on week 52		
Based on clients with high predictability	6.65%	**0.385**
Based on all clients	8.97%	0.222
Based on clients with low predictability	**6.35%**	0.370
With incremental learning; classes on week 52		
Based on clients with high predictability	4.95%	0.508
Based on all clients	5.38%	0.444
Based on clients with low predictability	**4.44%**	**0.548**

The first experiment case shows the perfect picture: predictions based on well-predicted clients are the most accurate, with predictions based on all clients in second place. The surprises arise when either incremental learning is used or clients in classes are chosen once.

The addition of incremental learning in the second experiment case results in a seemingly strange order, where clients with low predictability deliver better values than predictions based on well predicted clients. We suppose that because unpredictable clients' series have more noise, their time series are more similar to the ones of all clients. In addition, once the incremental model trains on examples of well predicted clients on week n, well predicted clients on week $n + 1$ may be different, and their time series may be different as well, which makes their scaled predictions look less smooth; unpredictable clients, on the other hand, have a higher chance to have a bigger overlap between subsequent weeks because the number of unpredictable clients on any week is always greater.

Another point to consider is that the client segmentation on the micro-level was completed based on binary forecast quality: whether there were any transactions in a

given day. The model on the meso- and macro-levels predicts the amount of spent money; therefore, even though the client is considered highly predictable on the micro-level (their sequence is full of *True*), they might have spent 100 money units on one day and 100,000 on the other, which makes the client a lot less predictable on other levels.

The last two cases show that predictions based on all clients perform poorly in comparison to predictions based on certain client classes on week 52. These outputs may be due to the decomposition of all clients' time series based on client predictability; forecasting distinct components is easier than the original series. Another interesting outcome is that the number of predictable clients on week 52 is 796 people out of 3000, so on the third experiment, 26.6% of the population can give more accurate predictions than when all clients are considered.

The difference between MAPE values for customers with high and low predictability for the first experiment case can be seen in Figure 8. Once the red dotted line goes under 0% MAPE, the predictions based on unpredictable clients perform better than the ones based on predictable clients. It can be observed that this happens during critical times: the lockdown announcement, the start of the war, and the time around New Year. The clients with high predictability show better results in quieter periods.
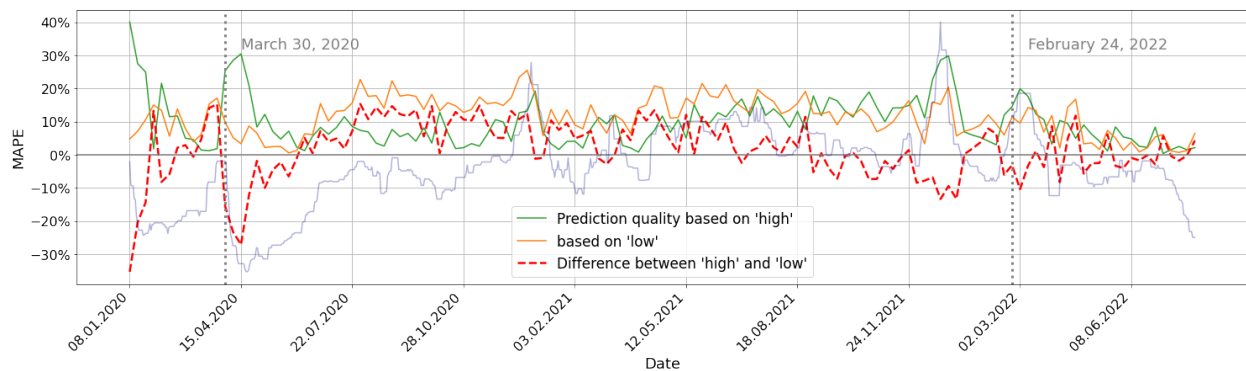

Figure 8. Difference between forecast quality based on clients with high and low predictability, where the pale blue line in the background depicts spent amount for all clients.

6. Conclusions

In our research, we developed the method for financial actors separation by predictability with the help of incremental learning on the micro-level, demonstrated its benefits and drawbacks for all clients forecasting on the macro-level, and showed how different predictability classes influence the model's forecast quality on the meso-level.

Our experiments show that the model with incremental learning was able to perform better on both the micro-levels in terms of F_1-score and meso- and macro-levels in terms of MAPE and hit probability, which supports our idea that incremental learning is useful for financial clients' behavior analysis considering the fact that clients predictability changes due to a lot of factors and events. Our research also shows that customer separation by predictability of their consumption behavior helps to improve the forecasting quality of the whole population's consumption due to the decomposition of all clients' time series, even though the best results are delivered by using the class of clients with low predictability. Further experiments with the same forecast objective for all levels are to be held in the future to ensure the same predictability class across levels.

The results are of practical value to companies operating in the finance sector that need to analyze their customers' behavior. Our research suggests that environmental instability such as pandemics or military action can be leveled out with incremental models, whereas the problem of huge data volumes and noise can be solved by client segmentation.

References

1. Koshkareva, M. Available online: https://github.com/Mpkosh/Enhancement_of_Consumption_Forecasting (accessed on 27 June 2023).
2. Dumitrescu, E.; Hué, S.; Hurlin, C.; Tokpavi, S. Machine learning for credit scoring: Improving logistic regression with non-linear decision-tree effects. *Eur. J. Oper. Res.* **2022**, *297*, 1178–1192. [CrossRef]
3. Karvana, K.G.M.; Yazid, S.; Syalim, A.; Mursanto, P. Customer Churn Analysis and Prediction Using Data Mining Models in Banking Industry. In Proceedings of the International Workshop on Big Data and Information Security (IWBIS), Bali, Indonesia, 11 October 2019; pp. 33–38. [CrossRef]
4. Al-Hashedi, K.G.; Magalingam, P. Financial fraud detection applying data mining techniques: A comprehensive review from 2009 to 2019. *Comput. Sci. Rev.* **2021**, *40*, 100402. [CrossRef]
5. Hernández-Nieves, E.; Hernández, G.; Gil-González, A.B.; Rodríguez-González, S.; Corchado, J.M. Fog computing architecture for personalized recommendation of banking products. *Expert Syst. Appl.* **2020**, *140*, 112900. [CrossRef]
6. Stavinova, E.; Bochenina, K.; Chunaev, P. *Predictability Classes for Forecasting Clients Behavior by Transactional Data*; Springer International Publishing: Berlin/Heidelberg, Germany, 2021; pp. 187–199. [CrossRef]
7. Bezbochina, A.; Stavinova, E.; Kovantsev, A.; Chunaev, P. Dynamic Classification of Bank Clients by the Predictability of Their Transactional Behavior. *Lect. Notes Comput. Sci.* **2022**, *13350*, 502–515.
8. Feng, Q.; Sun, X.; Hao, J.; Li, J. Predictability dynamics of multifactor-influenced installed capacity: A perspective of country clustering. *Energy* **2021**, *214*, 118831. [CrossRef]
9. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
10. Luo, Y.; Yin, L.; Bai, W.; Mao, K. An Appraisal of Incremental Learning Methods. *Entropy* **2020**, *22*, 1190. [CrossRef] [PubMed]
11. Gepperth, A.; Hammer, B. Incremental Learning Algorithms and Applications, In Proceedings of European Symposium on Artificial Neural Networks (ESANN), Bruges, Belgium, 27–29 April 2016.
12. Lu, J.; Liu, A.; Dong, F.; Gu, F.; Gama, J.; Zhang, G. Learning under Concept Drift: A Review. *IEEE Trans. Knowl. Data Eng.* **2019**, *31*, 2346–2363. [CrossRef]
13. van de Ven, G.M.; Tolias, A.S. Three scenarios for continual learning. *arXiv* **2019**, arXiv:1904.07734.
14. Lomonaco, V.; Maltoni, D. CORe50: A New Dataset and Benchmark for Continuous Object Recognition. *arXiv* **2017**, arXiv:1705.03550.
15. Matteoni, F.; Cossu, A.; Gallicchio, C.; Lomonaco, V.; Bacciu, D. Continual Learning for Human State Monitoring. *arXiv* **2022**, arXiv:2207.00010.
16. Rahman, M.S.; Wright, M.; Mandiant, S.E.C. On the Limitations of Continual Learning for Malware Classification. *arXiv* **2022**, arXiv:2208.06568.
17. Kovantsev, A. Probabilistic criteria for time-series predictability estimation. *Sci. Tech. J. Inf. Technol. Mech. Opt.* **2023**, 1–8. [CrossRef]